Perfomance Testing

What is it?

performance testing is a testing measure that evaluates the speed, responsiveness and stability of a computer, network, software program or device under a workload.

Why is it important?

Performance testing is important because it helps to ensure that a system can handle the expected workload and provides a good user experience. It can also help to identify bottlenecks and areas for optimization, and can help to prevent system failures or downtime.

How is it done?

Performance testing is typically done using a combination of automated and manual testing techniques. Automated testing tools can be used to simulate a large number of users or transactions, and to measure response times and throughput. Manual testing can be used to test the user interface and to evaluate the overall performance of the system under real-world conditions.

Prueba de Apache Benchmark

Pruebas de carga enviando múltiples solicitudes a un servidor web Apache para evaluar su rendimiento y capacidad de manejar la carga.

Resultados

```
(env) abrilcontreras@abrilcs:~/Documents/Abril/proyecto_db$ ab -n 1000 -c 50 http://127.0.0.1:8000/
This is ApacheBench, Version 2.3 <$Revision: 1913912 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking 127.0.0.1 (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Completed 700 requests
Completed 800 requests
Completed 900 requests
Completed 1000 requests
Finished 1000 requests


Server Software:        WSGIServer/0.2
Server Hostname:        127.0.0.1
Server Port:            8000

Document Path:          /
Document Length:        7997 bytes

Concurrency Level:      50
Time taken for tests:   4.203 seconds
Complete requests:      1000
Failed requests:        0
Total transferred:      8295000 bytes
HTML transferred:       7997000 bytes
Requests per second:    237.93 [#/sec] (mean)
Time per request:       210.145 [ms] (mean)
Time per request:       4.203 [ms] (mean, across all concurrent requests)
Transfer rate:          1927.38 [Kbytes/sec] received
```

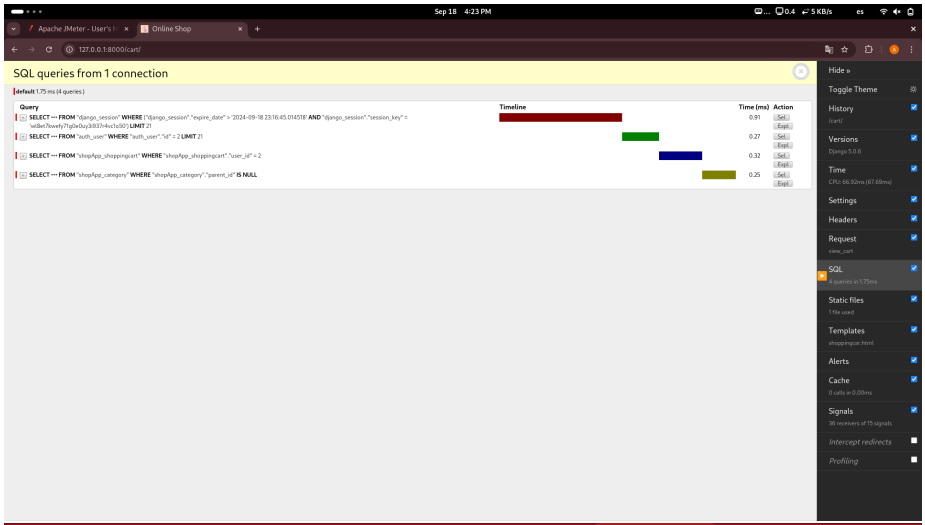
```
  Server Software:        WSGIServer/0.2
  Server Hostname:        127.0.0.1
  Server Port:            8000

  Document Path:          /
  Document Length:        7997 bytes

  Concurrency Level:      50
  Time taken for tests:   4.203 seconds
  Complete requests:      1000
  Failed requests:        0
  Total transferred:      8295000 bytes
  HTML transferred:       7997000 bytes
  Requests per second:    237.93 [#/sec] (mean)
  Time per request:       210.145 [ms] (mean)
  Time per request:       4.203 [ms] (mean, across all concurrent requests)
  Transfer rate:          1927.38 [Kbytes/sec] received

  Connection Times (ms)
                min  mean[+/-sd] median   max
  Connect:        0   76 269.1      0    1065
  Processing:    12  117 153.8     95    1759
  Waiting:        9   82 153.4     60    1731
  Total:         13  192 368.5     95    2799

  Percentage of the requests served within a certain time (ms)
    50%     95
    66%    105
    75%    115
    80%    124
    90%    168
    95%   1142
    98%   1374
    99%   1969
   100%   2799 (longest request)
(env) abrilcontreras@abrilcs:~/Documents/Abril/proyecto_db$
```

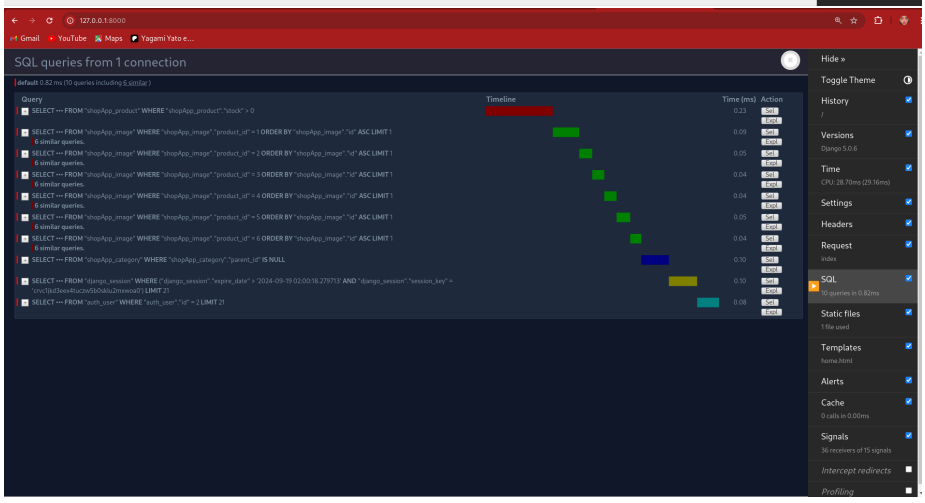
Prueba de Django Debug Toolbar

Django Debug Toolbar es una herramienta de desarrollo para Django que muestra información sobre el rendimiento de una aplicación web, como el tiempo de respuesta de la base de datos, el número de consultas realizadas y el tiempo de ejecución de cada vista.

Descargamos el django debug toolbar y lo instalamos en nuestro proyecto de django. Esto hará que aparezca en nuestro navegador una barra de herramientas con información sobre el rendimiento de nuestra aplicación.

Podemos ver el rendimiento de nuestra pagina como por ejemplo En el de SQL panel te muestra todas las consultas SQL que se han ejecutado en la página actual. Es muy útil para identificar consultas ineficientes o repetitivas.





La información de la barra Django Debug Toolbar indica que tu aplicación Django realizó 4 consultas SQL cuando se accedió a la página /cart/. Las consultas parecen estar relacionadas con la autenticación de usuario (auth_user), la tabla de sesiones de Django (django_session), el carrito de compras (shopApp_shoppingcart) y las categorías (shopApp_category).

Ddosify

Ddosify es una herramienta de prueba de carga y estrés de sitios web que permite a los usuarios simular un gran número de solicitudes a un servidor web para evaluar su rendimiento y capacidad de manejar la carga.

Podemos ver el cambio de rendimiento de nuestra pagina al agregar productos a la base de datos. En la primera imagen (Figure 2), la página se carga en aproximadamente 0.0302 segundos, mientras que en la segunda imagen (Figure 3), la página se carga en aproximadamente 0.0423 segundos. Esto puede deberse a que la página está realizando más consultas SQL o que está procesando más datos en la segunda imagen. Podemos entender que si tenemos más productos en nuestra base de datos, la pagina tardara mas en cargar.

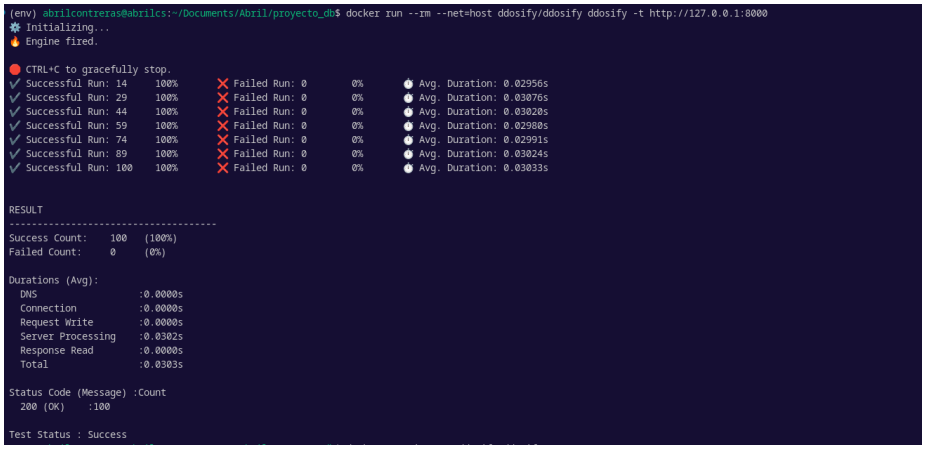


Figure 1: ddosify|Pagina de inicio sin productos

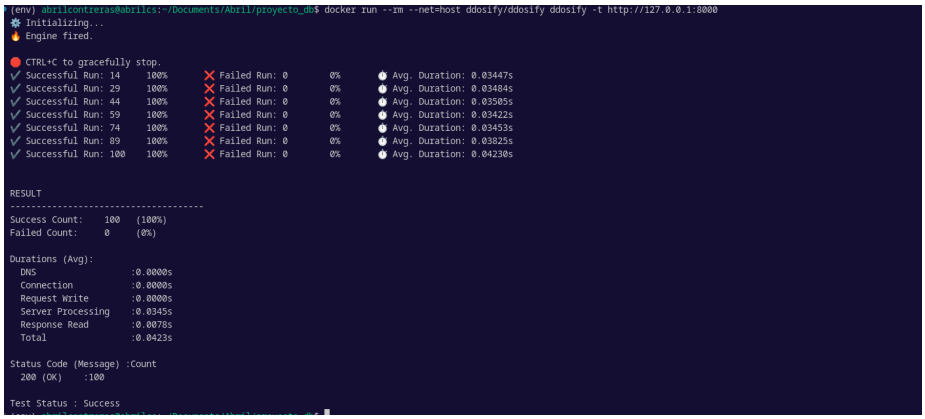


Figure 2: ddosify|Pagina de inicio con productos

El registro no tiene problemas.

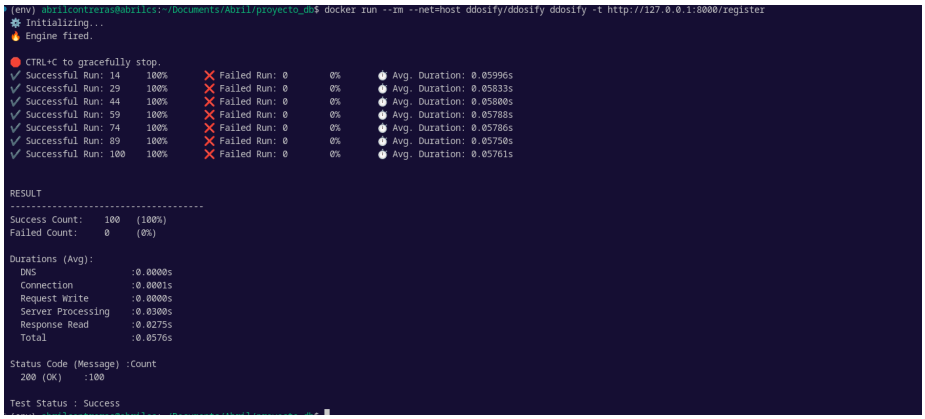


Figure 3: ddosify|Registro

Referencias

Gillis, A. S. (2023, 9 marzo). performance testing. Software Quality. https://www.techtarget.com/searchsoftwarequality/definition/performance-testing#:~:text= Performance%20testing%20is%20a%20testing, to%20identify%20performance%2Drelated%20bo